THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

Knowledge Distillation for Small-footprint Highway Networks

Citation for published version:
Lu, L, Guo, M & Renals, S 2017, Knowledge Distillation for Small-footprint Highway Networks. in *2017 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP 2017).* Institute of Electrical and Electronics Engineers (IEEE), pp. 4280-4284, 42nd IEEE International Conference on Acoustics, Speech and Signal Processing, New Orleans, Louisiana, United States, 5/03/17.
https://doi.org/10.1109/ICASSP.2017.7953072

Digital Object Identifier (DOI):
10.1109/ICASSP.2017.7953072

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
2017 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP 2017)

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

OPEN ACCESS

Download date: 08. May. 2023

KNOWLEDGE DISTILLATION FOR SMALL-FOOTPRINT HIGHWAY NETWORKS

Liang Lu[1] *Michelle Guo*[†2] *Steve Renals*[3]

[1]TTI-Chicago [2]Stanford University [3]The University of Edinburgh

llu@ttic.edu mguo95@stanford.edu s.renals@ed.ac.uk

arXiv:1608.00892v3 [cs.CL] 20 Dec 2016

ABSTRACT

Deep learning has significantly advanced state-of-the-art of speech recognition in the past few years. However, compared to conventional Gaussian mixture acoustic models, neural network models are usually much larger, and are therefore not very deployable in embedded devices. Previously, we investigated a compact highway deep neural network (HDNN) for acoustic modelling, which is a type of depth-gated feedforward neural network. We have shown that HDNN-based acoustic models can achieve comparable recognition accuracy with much smaller number of model parameters compared to plain deep neural network (DNN) acoustic models. In this paper, we push the boundary further by leveraging on the knowledge distillation technique that is also known as *teacher-student* training, i.e., we train the compact HDNN model with the supervision of a high accuracy cumbersome model. Furthermore, we also investigate sequence training and adaptation in the context of teacher-student training. Our experiments were performed on the AMI meeting speech recognition corpus. With this technique, we significantly improved the recognition accuracy of the HDNN acoustic model with less than 0.8 million parameters, and narrowed the gap between this model and the plain DNN with 30 million parameters.

Index Terms— Knowledge distillation, Highway deep neural networks, Small-footprint

1. INTRODUCTION

Recent years have witnessed wide applications of speech technology in embedded devices like mobile phones, thanks to deep learning that has significantly advanced state-of-the-art in this area. For scenarios that internet connections are unavailable or for privacy concerns, it is desirable that speech recognisers can run locally in such kind of resource constrained platforms. However, state-of-the-art neural network models are either computationally expensive or consume large amount of memory, and are therefore unsuitable for this purpose. Recently, there have been a number of works on small footprint acoustic models to address this problem such as using low-rank matrices [1, 2], structured linear layers [3, 4, 5], and the use of low rank displacement of structured matrices [4]. Instead of manipulating the model parameters, another approach is based on the *teacher-student* architecture [6, 7, 8], which is also known as model compression [9] or knowledge distillation [10]. In this approach, the *teacher* may be a large-size network or an ensemble of several different models, which is used to predict the soft targets for training the *student* model that is much smaller. As pointed out in [10], the soft targets provided by the teacher encode the generalisation power of the teacher, and the student model trained using these pseudo labels is observed to perform better than the same model trained independently using the ground truth labels [10].

Previously, we studied a compact acoustic model using highway deep neural network (HDNN) for resource constrained speech recognition [11]. HDNN is a type of network with shortcut connections between hidden layers [12]. Compared to the plain networks with skip connections, HDNNs are equipped with two gate functions – *transform* and *carry* gate – to control and facilitate the information flow over all the whole network. In particular, the transform gate is used to scale the output of a hidden layer and the carry gate is used to pass through the input directly after elementwise rescaling. The gate functions are the key to train very deep networks [12] and to speed up convergence as experimentally validated in [11]. We have shown that the gate functions can manipulate the behavior of the whole neural networks in sequence training and adaptation [13]. With the gate functions, we can train much thinner and deeper networks with much smaller number of model parameters, which can achieve comparable recognition accuracy compared to much larger plain DNNs.

In this paper, we investigate teacher-student training to further improve the accuracy of the small-footprint HDNN acoustic model. In particular, we use a large size plain DNN acoustic model to provide soft labels for training the student HDNN model. As mentioned before, there have been a number of work on teacher-student training for speech recognition [6, 7, 10]. The one that is closest to our study is [6]. However, the student model investigated in this paper is much smaller due to the highway architecture. In addition, we present further analysis and experimental study on hybrid loss functions that interpolate the cross-entropy and teacher-student costs, the use of temperature to smooth the soft labels as well as sequence training and adaptation results in this context.

2. HIGHWAY DEEP NEURAL NETWORKS

In this work, we focus on feed-forward neural networks – also known as DNNs – as target student models. Although long short-term memory based recurrent neural networks (LSTM-RNNs) and convolutional neural networks (CNNs) may obtain higher recognition accuracy compared to DNNs [14, 15], they are more suitable for teacher models because they are usually computationally more expensive for applications on resource constrained platforms. A plain DNN with L hidden layers may be represented as

$$\boldsymbol{h}_t^{(1)} = \sigma(\boldsymbol{x}_t, \theta_1) \tag{1}$$

$$\boldsymbol{h}_t^{(l)} = \sigma(\boldsymbol{h}_t^{(l-1)}, \theta_l), \quad \text{for} \quad l = 2, \ldots, L \tag{2}$$

$$\boldsymbol{y}_t = g(\boldsymbol{h}_t^{(L)}, \varphi) \tag{3}$$

where $\boldsymbol{x}_t$ is an input vector to the network at the time step t; $\sigma(\boldsymbol{h}_t^{(l-1)}, \theta_l)$ denotes the transformation of the input $\boldsymbol{h}_t^{(l-1)}$ with the parameter θ_l followed by a nonlinear activation function, e.g.,

[†]The work was partially done when M. Guo was visiting The University of Edinburgh under the Stanford Bing Overseas Studies Programme.

sigmoid; $g(\cdot, \varphi)$ is the output function that is parameterised by φ in the output layer, which usually uses the softmax to obtain the posterior probability of each class given the input feature.

Highway deep neural networks [12] augment the feature extractor with gate functions, in which the hidden layer may be represented as

$$\begin{aligned} \boldsymbol{h}_t^{(l)} = \sigma(\boldsymbol{h}_t^{(l-1)}, \theta_l) \circ T(\boldsymbol{h}_t^{(l-1)}, \boldsymbol{W}_T) \\ + \boldsymbol{h}_t^{(l-1)} \circ C(\boldsymbol{h}_t^{(l-1)}, \boldsymbol{W}_c), \end{aligned} \tag{4}$$

where $T(\cdot)$ is the *transform* gate that scales the original hidden activations; $C(\cdot)$ is the *carry* gate, which scales the input before passing it directly to the next hidden layer; $\circ$ denotes elementwise multiplication; The outputs of $T(\cdot)$ and $C(\cdot)$ are constrained to be within $[0, 1]$, and we use the sigmoid function for both gates that are parameterised by $\mathbf{W}_T$ and $\mathbf{W}_c$ respectively. Following our previous work [12], we tie the parameters in the gate functions across all the hidden layers, which can significantly save model parameters. In this work, we do not use any bias vector in the two gate functions. As pointed out in [13], to speed up the training, we can pack the matrices as

$$\tilde{\boldsymbol{W}}_l = \left[\boldsymbol{W}_l^\top, \boldsymbol{W}_T^\top, \boldsymbol{W}_c^\top\right]^\top, \tag{5}$$

where $\boldsymbol{W}_l^\top$ is the weight matrix in the l-th layer, and we then compute $\tilde{\boldsymbol{W}}_l \boldsymbol{h}_{l-1}$ once for all. By this trick, we can leverage on the power of GPUs on computing large matrix-matrix multiplications efficiently in the minibatch mode.

3. MODEL TRAINING

3.1. Cross-Entropy Training

The most common criterion to train neural networks for classification is cross-entropy (CE), which defines the loss function as

$$\mathcal{L}^{(CE)}(\theta) = -\sum_j \hat{y}_{jt} \log y_{jt}, \tag{6}$$

where j is the index of the hidden Markov model (HMM) state; $\boldsymbol{y}_t$ is the output of the nerual network as Eq. (3), while $\hat{\boldsymbol{y}}_t$ denotes the ground truth label that is a one-hot vector. Note that, the loss function is defined with one training utterance here for the simplicity of notation. Supposing that $\hat{y}_{jt} = \delta_{ij}$, where δ_{ij} is the Kronecker delta function and i is the ground truth class at the time step t, the CE loss becomes

$$\mathcal{L}^{(CE)}(\theta) = -\log y_{it}. \tag{7}$$

In this case, minimising $\mathcal{L}^{(CE)}(\theta)$ is equivalently to minimise the negative log posterior probability of the correct class, and it is equal to maximising the probability y_{it}, while the posterior probabilities of other classes are ignored. However, maximising y_{it} will also result in minimising the posterior probabilities of other classes since they sum to one.

3.2. Teacher-Student Training

Instead of using the ground truth labels, the teacher-student training approach defines the loss function as

$$\mathcal{L}^{(KD)}(\theta) = -\sum_j \tilde{y}_{jt} \log y_{jt}, \tag{8}$$

where $\tilde{y}_{jt}$ is the output of the teacher model, which works as a pseudo label. As pointed out in [6], the loss function as Eq. (8) is equivalent to minimise the Kullback-Leibler divergence between the posterior probabilities of each class from the teacher and student models. Here, $\tilde{y}_{jt}$ is no longer a one-hot vector, instead, the competing classes will have small but nonzero posterior probabilities for each training example. Hinton et al. [10] suggested that the small posterior probabilities are valuable information that encodes correlations among different classes. However, their roles may be very small in the loss function as these probabilities are close to zero due to the softmax function. To address this problem, they suggested to use a large temperature to flatten the posterior distribution as

$$y_{jt} = \frac{\exp\left(z_{jt}/T\right)}{\sum_i \exp\left(z_{it}/T\right)}, \tag{9}$$

$$\boldsymbol{z}_t = \boldsymbol{W}^{(L+1)} \boldsymbol{h}_t^{(L)} + \boldsymbol{b}^{(L+1)}, \tag{10}$$

where $\boldsymbol{W}^{(L+1)}, \boldsymbol{b}^{(L+1)}$ are parameters in the softmax layer, and $T \in \mathbb{R}^+$ is the temperature. Following [10], we applied the same temperature to the softmax functions in both the teacher and student networks in our experiments, as only increasing the temperature in the teacher network resulted in much higher error rates in our experiments.

One particular advantage of the teacher-student training approach is that unlabelled data can be used easily. However, when the ground truth labels are available, it may be beneficial to incorporate the ground truth information into the loss function, which can be done by interpolating the two loss functions as

$$\widetilde{\mathcal{L}(\theta)} = \mathcal{L}^{(KD)}(\theta) + q\mathcal{L}^{(CE)}(\theta) \tag{11}$$

where $q \in \mathbb{R}^+$ is the tuning parameter. We denote this as the hybrid loss, and it will be studied in the experimental section.

3.3. Sequence Training

While the previous two loss functions are defined at the frame level, sequence training defines the loss at the sequence level, which usually yields significant improvement for speech recognition [16, 17, 18]. If we denote $\boldsymbol{X}$ as the sequence of acoustic frames $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$ and $\boldsymbol{Y}$ as the sequence of labels, where T is the length of the signal, the loss function from the state-level minimum Bayesian risk criterion(sMBR) [19, 20] is defined as

$$\mathcal{L}^{(sMBR)}(\theta) = \frac{\sum_{\mathcal{W}\in\Phi} p(\boldsymbol{X} \mid \mathcal{W})^k P(\mathcal{W}) A(\boldsymbol{Y}, \hat{\boldsymbol{Y}})}{\sum_{\mathcal{W}\in\Phi} p(\boldsymbol{X} \mid \mathcal{W})^k P(\mathcal{W})}, \tag{12}$$

where $A(\boldsymbol{Y}, \hat{\boldsymbol{Y}})$ measures the state level distance between the ground truth and predicted labels; Φ denotes the hypothesis space represented by a denominator lattice, and $\mathcal{W}$ is the word-level transcription; k is the acoustic score scaling parameter. In this paper, we only focus on the sMBR criterion since it can achieve comparable or slightly better results compared to the maximum mutual information (MMI) or minimum pone error (MPE) criterion [17].

For sequence training, the acoustic model is normally firstly trained with the CE loss function, which is then fine tuned with the sequence-level loss for a few iterations. While for knowledge distillation, the model is firstly trained with the loss function as Eq. (8). This may raise the question that if the improvement will diminish in sequence training, and we will perform experimental study to answear this question. Note that, only applying the sequence training criterion without regularisation may lead to overfitting as observed

Table 1. Baseline results of DNN and HDNN systems with CE and sMBR training. The DNN systems were built using Kaldi toolkit, where the networks were pre-trained using restricted Bolzman machines. Results are shown in terms of word error rates (WERs). We use H to denote the size of hidden units, and L the number of layers.

		eval		dev	
Model	Size	CE	sMBR	CE	sMBR
DNN-$H_{2048}L_6$	$30M$	26.8	24.6	26.0	24.3
DNN-$H_{512}L_{10}$	$4.6M$	28.0	25.6	26.8	25.1
DNN-$H_{256}L_{10}$	$1.7M$	30.4	27.5	28.4	26.5
DNN-$H_{128}L_{10}$	$0.71M$	34.1	30.8	31.5	29.3
HDNN-$H_{512}L_{10}$	$5.1M$	27.2	24.9	26.0	24.5
HDNN-$H_{256}L_{10}$	$1.8M$	28.6	26.0	27.2	25.2
HDNN-$H_{128}L_{10}$	$0.74M$	32.0	29.4	29.4	28.1
HDNN-$H_{512}L_{15}$	$6.4M$	27.1	24.7	25.8	24.3
HDNN-$H_{256}L_{15}$	$2.1M$	28.4	25.9	26.9	25.2

in [17, 18]. To address this problem, we interpolate the sMBR loss function with the CE loss [18]. However, for the case of knowledge distillation, we apply the following interpolation:

$$\widehat{\mathcal{L}(\theta)} = \mathcal{L}^{(sMBR)}(\theta) + p\mathcal{L}^{(KD)}(\theta), \quad (13)$$

where $p \in \mathbb{R}^+$ is the smoothing parameter.

4. EXPERIMENTS

4.1. System Setup

Our experiments were performed on the individual headset microphone (IHM) subset of the AMI meeting speech transcription corpus [21].The amount of training data is around 80 hours, corresponding to roughly 28 million frames. We followed the experimental setup in [13]. We used 40-dimensional fMLLR adapted features vectors normalised on the per-speaker level, which were then spliced by a context window of 15 frames (i.e. ± 7). The number of tied HMM states is 3927. The HDNN models were trained using the CNTK toolkit [22], while the results were obtained using the Kaldi decoder [23]. We also used the Kaldi tookit to compute the alignment and lattices for sequence training. We set the momentum to be 0.9 after the 1st epoch for CE training, and we used the sigmoid activation for all the networks. The weights in each hidden layer of HDNNs were randomly initialised with a uniform distribution in the range of $[-0.5, 0.5]$ and the bias parameters were initialised to be 0 for CNTK systems. We used a trigram language model for decoding. The word error rates (WERs) of the baseline systems with different model structures are shown in Table 1.

4.2. Loss Function and Temperature

We firstly compare the teacher-student loss function as Eq. (8) and the hybrid loss function as Eq. (11). We used a CE trained plain DNN-$H_{2048}L_6$ as the teacher model, and used the HDNN-$H_{128}L_{10}$ as the student model. Figure 1 shows the convergence curves when training the model with different loss functions, while Table 2 shows the WERs. We observe that by teacher-student training without the ground truth labels, we can achieve significantly lower frame error rate on the cross validation set as shown in Figure 1, corresponding to moderate WER reduction (31.3% vs. 32.0 on the `eval` set). However, using the hybrid loss function as Eq. (11), we do not obtain further improvement. In fact, it converges slower when $q > 0$

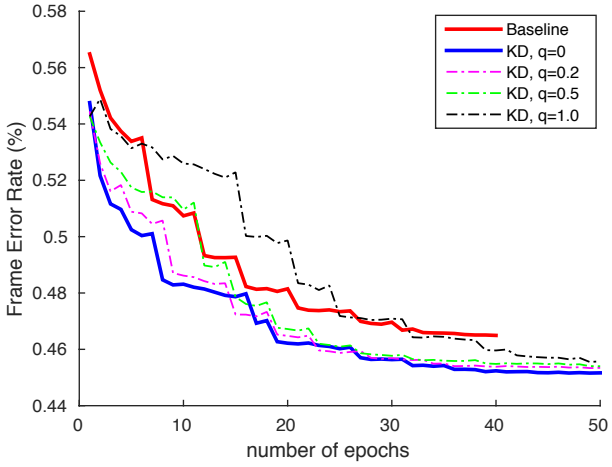

Fig. 1. Convergence curves of teacher-student training. The frame error rates were obtained from the cross validation set. It slows down the convergence as q increases.

Table 2. Results of teacher-student training with different loss functions and temperatures.

			WER	
Model	q	T	eval	dev
DNN-$H_{128}L_{10}$	–	–	34.1	31.5
HDNN-$H_{128}L_{10}$ baseline	–	–	32.0	29.9
HDNN-$H_{128}L_{10}$	0	1	31.3	29.3
HDNN-$H_{128}L_{10}$	0.2	1	31.4	29.5
HDNN-$H_{128}L_{10}$	0.5	1	31.3	29.4
HDNN-$H_{128}L_{10}$	1.0	1	31.3	29.4
HDNN-$H_{128}L_{10}$	0	2	32.3	29.9
HDNN-$H_{128}L_{10}$	0	3	33.0	30.6

during training as shown in Figure 1. Our interpretation is that it may be because the probabilities of uncorrected classes played a smaller role in this case, which supports the argument that they encode useful information for training the student model [10]. This hypothesis encouraged us to investigate the use of a large temperature to flatten the posterior probability distribution of the labels from the teacher model. The results are also shown in Table 2. Contrary to our expectation, using large temperatures results in higher WERs. In the following experiments, we fixed $q = 0$ and $T = 1$.

4.3. Teacher Model

We then improved the teacher model by sMBR-based sequence training, and used this model to supervise the training of the student model. Similar to the observations in [6], the sMBR-based teacher model can significantly improve the performance of the student model. In fact, the error rate is lower than that achieved by the student model trained independently with sMBR as shown in Table 3 (28.8% vs. 29.4% on the `eval` set). Note that, since the sequence training criterion is not to maximise the frame accuracy, training the model with this criterion normally reduces the frame accuracy, as shown explicitly by Figure 6 in [24]. Interestingly, we observed the same pattern in the case of teacher-student training. Figure 2 shows the convergence curves of using CE and sMBR based teacher models, where we see that the student model achieves much higher frame error rate on the cross validation set when supervised by sMBR-

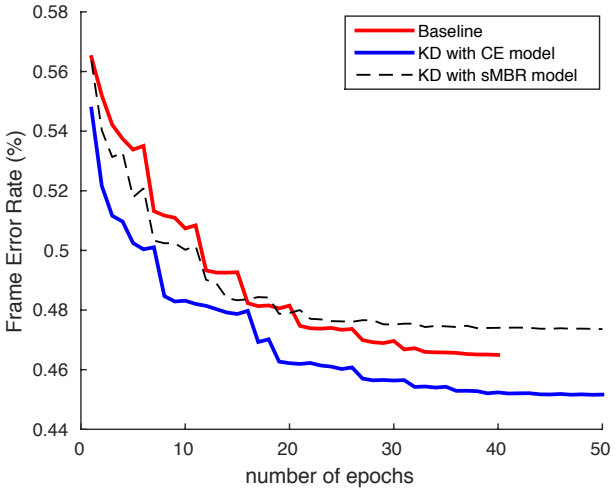

Fig. 2. Convergence curves of teacher-student training with CE or sMBR-based teacher model.

Table 3. Results of sequence training on the `eval` set for the student HDNN model. LR denotes the learning rate.

Teacher	LR	p	$\mathcal{L}^{(KD)} \rightarrow \widehat{\mathcal{L}(\theta)}$
DNN-$H_{2048}L_6$-CE	1×10^{-5}	0.2	31.3 $\rightarrow$ 28.4
DNN-$H_{2048}L_6$-sMBR	1×10^{-5}	0.2	28.8 $\rightarrow$ 28.9
DNN-$H_{2048}L_6$-sMBR	1×10^{-5}	0.5	28.8 $\rightarrow$ 28.0
DNN-$H_{2048}L_6$-sMBR	5×10^{-6}	0.2	28.8 $\rightarrow$ 28.6
DNN-$H_{2048}L_6$-sMBR	5×10^{-6}	0.5	28.8 $\rightarrow$ 28.0

based teacher model, though the loss function as Eq. (8) is at the frame level.

We then study if the accuracy of the student model can be further improved by the sequence level criterion. Here, we set the smoothing parameter $p = 0.2$ in Eq. (13) and the default learning rate to be 1×10^{-5} following our previous setup in [13]. Table 3 shows the sequence training results of student models supervised by the CE and sMBR-based teacher models respectively. Not surprisingly, the student model supervised by the CE-based DNN model can be significantly improved by the sequence training. Notably, the WER obtained by this approach is lower compared to the model trained independently with sMBR (28.4% vs. 29.4% on the `eval` set). However, this configuration did not work for the student model supervised by the sMBR-based teacher model. After inspection, we found that it was due to overfitting. We then increased the value of p for stronger regularisation and reduced the learning rate. Lower WERs can be obtained as the table shows, however, the improvement is less significant as the sequence level information has already been integrated into the teacher model.

4.4. Unsupervised Adaptation

Our final experiments concern adaptation. Neural network acoustic models are less adaptable due to large number of unstructured model parameters compared to conventional acoustic models using Gaussian mixtures. However, a smaller model may be easier to adapt. In particular, the gate functions in HDNNs are more adaptable since they have much smaller number of model parameters, e.g., the total number of parameters in $(\boldsymbol{W}_T, \boldsymbol{W}_C)$ of an HDNN-$H_{128}L_{10}$ acoustic model is around 0.03 million. Furthermore, only updating the gate functions does not easily yield overfitting with small amount of adaptation data and pseudo labels [13]. We performed similar adaptation experiments for HDNN trained by the teacher-student approach. We applied the second-pass adaptation approach for the standalone HDNN model, i.e., we decoded the evaluation utterances to obtain the hard labels first, and then used these labels to adapt the model using the CE loss as Eq. (7). However, using the teacher-student loss as Eq. (8), only one pass decoding is required because the pseudo labels for adaptation are provided by the teacher model, which does not need the word level transcription. This is a particular advantage of the teacher-student training technique. However, note that for resource constrained application scenarios, the student model should be adapted offline, because otherwise the teacher model needs to be accessed to generate the labels. This requires another set of unlabelled speaker-dependent data for adaptation, but it is usually not expensive to collect.

Table 4. Results of unsupervised speaker adaptation. DP denotes the number of decoding passes.

				`eval`	
Model	Loss	Update	DP	SI	SD
HDNN-$H_{128}L_{10}$	$\mathcal{L}^{CE}$	All	2	29.4	28.8
HDNN-$H_{128}L_{10}$	$\mathcal{L}^{CE}$	Gates	2	29.4	28.7
HDNN-$H_{128}L_{10}$-KD	$\mathcal{L}^{KD}$	All	1	28.4	27.5
HDNN-$H_{128}L_{10}$-KD	$\mathcal{L}^{KD}$	Gates	1	28.4	27.8
HDNN-$H_{128}L_{10}$-KD	$\mathcal{L}^{CE}$	All	2	28.4	27.7
HDNN-$H_{128}L_{10}$-KD	$\mathcal{L}^{CE}$	Gates	2	28.4	27.1

Since the standard AMI corpus does not have this additional set of speaker-dependent data, we only show online adaptation results. We used the teacher-student trained model from row 1 of Table 3 as the speaker-independent (SI) model because its pipeline is much simpler. The baseline system used the same network as the SI model, but it was trained independently. During adaptation, we updated the SI model by 5 iterations with fixed learning rate as 2×10^{-4} per sample following our previous setup [13]. We also compared the CE loss as Eq. (7) and the teacher-student loss as Eq. (8) for adaptation. Results are given in Table 4. Using the CE loss function for both SI models, only updating the gates yields slightly better results, while updating all the model parameters gives smaller improvements, possibly due to overfitting. Interestingly, this is not the case for the teacher-student loss, i.e. updating all the model parameters yields lower WER. These results may agree with the argument in [10] that the soft targets can work as a regulariser and can prevent the student model from overfitting.

5. CONCLUSIONS

In this paper, we investigated the teacher-student training for small-footprint acoustic models using HDNNs. We observed that the accuracy of the student acoustic model could be improved under the supervision of a high accuracy teacher model, even without additional unsupervised data. In particular, the student model supervised by a sMBR-based teacher model achieved lower WER compared to the model trained independently using the sMBR-based sequence training approach. Unsupervised speaker adaptation further improved the recognition accuracy by around 5% relative for our model with less then 0.8 million model parameters. However, we did not obtain improvements by using the hybrid loss function by interpolating the CE and teacher-student loss functions, and using higher temperature to smooth the pseudo labels did not help either. In the future, we shall evaluate this model on low resource conditions where the amount of training data is much smaller.

6. REFERENCES

[1] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition." in *Proc. INTERSPEECH*, 2013, pp. 2365–2369.

[2] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *Proc. ICASSP*. IEEE, 2013, pp. 6655–6659.

[3] Q. Le, T. Sarlós, and A. Smola, "Fastfood-approximating kernel expansions in loglinear time," in *Proc. ICML*, 2013.

[4] V. Sindhwani, T. N. Sainath, and S. Kumar, "Structured transforms for small-footprint deep learning," in *Proc. NIPS*, 2015.

[5] M. Moczulski, M. Denil, J. Appleyard, and N. de Freitas, "ACDC: A Structured Efficient Linear Layer," in *Proc. ICLR*, 2016.

[6] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning small-size DNN with output-distribution-based criteria," in *Proc. INTERSPEECH*, 2014.

[7] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Proc. NIPS*, 2014, pp. 2654–2662.

[8] R. Adriana, B. Nicolas, K. Samira Ebrahimi, C. Antoine, G. Carlo, and B. Yoshua, "Fitnets: Hints for thin deep nets," in *Proc. ICLR*, 2015.

[9] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. ACM SIGKDD*, 2006.

[10] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS Deep Learning and Representation Learning Workshop*, 2015.

[11] L. Lu and S. Renals, "Small-footprint deep neural networks with highway connections for speech recognition," in *Proc. INTERSPEECH*, 2016.

[12] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Proc. NIPS*, 2015.

[13] L. Lu, "Sequence training and adaptation of highway deep neural networks," in *Proc. SLT*, 2016.

[14] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." in *Proc. INTERSPEECH*, 2014.

[15] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 10, pp. 1533–1545, 2014.

[16] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum bayes risk training of deep neural network acoustic models using distributed hessian-free optimization," in *Proc. INTERSPEECH*, 2012.

[17] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. INTERSPEECH*, 2013.

[18] H. Su, G. Li, D. Yu, and F. Seide, "Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription," in *Proc. ICASSP*. IEEE, 2013, pp. 6664–6668.

[19] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition." in *Proc. INTERSPEECH*. Citeseer, 2006.

[20] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Proc. ICASSP*. IEEE, 2009, pp. 3761–3764.

[21] S. Renals, T. Hain, and H. Bourlard, "Recognition and understanding of meetings the AMI and AMIDA projects," in *Proc. ASRU*. IEEE, 2007, pp. 238–247.

[22] D. Yu, A. Eversole, M. Seltzer, K. Yao, Z. Huang, B. Guenter, O. Kuchaiev, Y. Zhang, F. Seide, H. Wang *et al.*, "An introduction to computational networks and the computational network toolkit," Tech. Rep. MSR, Microsoft Research, Tech. Rep., 2014.

[23] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlıcek, Y. Qian, P. Schwarz, J. Silovský, G. Semmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[24] G. Heigold, E. McDermott, V. Vanhoucke, A. Senior, and M. Bacchiani, "Asynchronous stochastic optimization for sequence training of deep neural networks," in *Proc. ICASSP*. IEEE, 2014, pp. 5587–5591.